REMARKS

This Amendment is responsive to the Final Official Action mailed on January 25, 2005, for which a two-month extension is hereby requested, and is being filed concurrently with a Request for Continued Examination. The Office Action objected to claim 33 and rejected the other pending claims, with claims 25 and 34-37 rejected under 35 U.S.C. 112, first paragraph, claims 9, 11, and 12 rejected under 35 U.S.C. 102(b) as anticipated by Kojima *et al.* (U.S. patent number 5,880,981), claims 12, 25, 27, 29, and 39 rejected under 35 U.S.C. 102(e) as anticipated by Pickett *et al.* (U.S. patent number 6,101,595), claims 9, 25, 26, 29-32, 39, and 40 rejected under 35 U.S.C. 102(b) as anticipated by Yamada (U.S. patent number 5,765,212), and claims 28 and 38 rejected under 35 U.S.C. 102(b) as anticipated by Suzuki *et al.* (U.S. patent number 5, 809,306). Claim 33 has been rewritten in independent form to conform to the remarks of the Office Action and should consequently now be allowable. Claims 9 and 12 have been cancelled, new claims 41 and 42 have been added, and claims 10, 11, 25, 28, 31, 34, and 38 have been amended. For the reasons stated below all of the pending claims are now believed to be allowable.

Telephone interview

The Examiner is thanked for the telephone interview of May 3, 2005, with the undersigned concerning the use of the term "contiguous" and on whether it implied lack of gaps. It was decided that the most reasonable course was just to make this explicit in the claims. More specifically, independent claims 10, 11, 25, and 34 all now specify all of a select group of instruction is stored "contiguously without any gaps".

Allowable subject matter and new claims

The Office Action indicated that claim 33 would be allowable if rewritten in independent form that incorporated all underlying claims. This has been done and claim 33 should now be allowable. Claims 41 and 42 have been added. These new claims are both dependent claims that include limitations previously found in other dependent claims, but which have claim 33 as their base claim. Consequently, claims 41 and 42 should also be allowable.

Rejections under 35 U.S.C. 112, first paragraph, and claim amendments

The Office Action rejected claims 25 and 34-37 under 35 U.S.C. 112, first paragraph, related to the language of "all of the members of the instruction set" in claim 34 and of "the instruction set" in claim 25. These claims have been amended along the lines suggested in the Office Action to make more clear that what is being specified is "a selected set of instructions of the instruction set" and not necessarily the whole of the instruction set. Therefore, it is respectfully submitted that rejection of claims 25 and 34, along with dependent claims 35-37, under 35 U.S.C. 112, first paragraph, is not well founded and should be withdrawn.

Claims 28 and 38 have also been amended for consistency with their underlying base claim, claim 25.

Claims 10 and 11, which have now been rewritten in independent form by incorporating 9, have also had similar language incorporated. Claim 31 has had its dependence changed from now cancelled claim 9 to claim 10.

Rejections under 35 U.S.C. 102(b) based on Kojima *et al.*

The Office Action rejected claim 11 under 35 U.S.C. 102(b) as anticipated by Kojima *et al.*. Claim 11 recites that a memory that is a one time programmable (OTP) memory, which is discussed at page 1, line 21 *et seq.*, page 4, line 30 *et seq.*, and elsewhere in the present application. The Office Action cites element 5 of Figure 1 and column 4, lines 23-24, of Kojima. In this location, Kojima states this memory "may be composed of read only memory (ROM), random access memory (RAM), and preferably a combination of both." Both ROM and RAM are distinct from a one time programmable (OTP) memory: a one time programmable memory allows a user to program it once, but can then not be overwritten. Thus, a one time programmable memory differs from both a ROM, which does not allow the user to program into it at all (having been written before the user has it), and a RAM, which allows its contents to be overwritten. Although more generally applicable, the present invention is particularly applicable to the case of a one time programmable memory and a one time programmable memory is particularly suited to the exemplary embodiments of the present invention. No specific disclosure of the use of one time programmable (OTP) memories can be found at cited location (or elsewhere) in Kojima. Consequently, a rejection of claim 11 under 35 U.S.C. 102(b) is respectfully submitted to be in error and should be withdrawn.

Claim 11 additionally recites a "a memory for storing the instructions, wherein the instructions are stored contiguously without any gaps". The Office Action refers to element 5 of Figure 1 and column 4, lines 23-24, of Kojima. However, no specific disclosure of the instructions being stored contiguously can be found at this location (or elsewhere) in the reference, only the description of a 24-bit wide ROM for storing 3-byte (and only 3-byte) instructions. Consequently, a rejection of claim 11 under 35 U.S.C. 102(b) is respectfully submitted to be in error and claim 11 is believed further allowable for this reason.

Rejections under 35 U.S.C. 102(e) based on Pickett *et al.*

The Office Action rejected claims 10, 25, 27, 29, and 39 under 35 U.S.C. 102(e) as anticipated by Pickett *et al.* This rejection is also respectfully submitted to be in error as Pickett neither discloses nor suggests elements found in these claims. In particular, independent claims 10 and 25 respectively contain the limitations (as amended) of "a memory for storing the instructions, wherein all of a selected set of the instructions is stored contiguously without any gaps" and "all of the selected set of instructions is stored contiguously without any gaps in the memory", where, in accordance with the remarks in the "Response to Arguments" section of the Office Action and as discussed in the interview, the "without any gaps" is now explicitly found in the claims. Pickett describes all of the bytes of a single given instruction as being stored contiguously, and also describes *some* instances of separate instructions as being stored contiguously without any gaps, but Pickett neither teaches nor suggests that "*all* of a selected set of the instructions ... stored contiguously without any gaps"[emphasis added] in the memory; rather, Pickett explicitly shows there to be gaps in the storage instructions.

More specifically, second element of claims 10 and 25 respectively read

> a memory for storing the instructions, wherein *all of a selected set* of the instructions *is stored contiguously without any gaps*

and

> programming a selected set of instructions of the instruction set into the memory, wherein *all of the selected set of instructions is stored contiguously without any gaps* in the memory

where the emphasis is added. The Office Action identifies the memory with element 112 ("Instruction Storage") in Figure 3 of Pickett and notes that in Figure 5 instructions IN2 and IN3 are contiguous. However, these are only two specific elements of the stored instructions: Pickett neither teaches nor suggests that "*all* of the selected set of instructions is stored

contiguously without any gaps in the memory". Rather, Pickett explicitly shows gaps between some of the instructions in Figure 5.

This lack of contiguous storage is shown explicitly in Pickett's Figure 5, which shows the organization of an exemplary cache line. Figure 5 shows a number of instructions (IN0, IN1, ...) taken from the instruction set. It also shows a number of predicted branch indicators (PB0, PB1). As described beginning at column 16, line 57, these predicted branch indicators *are not part of the instruction set*, but, rather, are generated by the system based upon the preceding instruction in the cache line. As shown in Figure 5, when the cache line is formed up, these are interposed between instructions from the instruction set; for example, PB0 is placed between IN1 and IN2. Thus, there is a gap between the instructions IN1 and IN2 that does not contain an element of the instruction set. Consequently, for the cited element 112, Pickett explicitly *teaches away* from "programming a selected set of instructions of the instruction set into the memory, wherein all of the selected set of instructions is stored contiguously without any gaps in the memory".

(Additionally, although Pickett uses the term "contiguous", it should be noted that the use of "contiguous" in Pickett has a very particular meaning. This is described at column 7, lines 8-11: "As used herein, the term 'group of contiguous instruction bytes' is used to refer to the instruction bytes which are provided by the instruction cache in a particular clock cycle in response to a fetch address." This is not the same as *storing* all of the selected members of an instruction set contiguously without gap, meaning in an adjacent manner without gaps, *within* a memory. When Pickett uses "contiguous", this only describes the manner in which instruction bytes as supplied from the memory, while the claims relate to placement of the instruction set within the physical memory. Further, Pickett is only referring to those instructions provided in a given fetch, whereas the claims describe how "all of a selected set of the instructions is stored".)

Therefore, for any on the preceding reasons, it is respectfully submitted that a rejection of claim 10 and claim 25, along with dependent claims 27, 29, and 39, under 35 U.S.C. 102(e) as anticipated by Pickett is not well founded and should be withdrawn.

Additionally, claims 31 and 32 have claim 10 as their base claim and are believed allowable on this basis.

Rejections under 35 U.S.C. 102(b) based Yamada

The Office Action rejected claims 25, 26, 29-32, 39, and 40 under 35 U.S.C. 102(b) as anticipated by Yamada. It is believed that the rejection of independent claim 25 on this basis is not well founded. Claims 26, 29, 30, 39, and 40 all depend on claim 25. Concerning claims 31 and 32, these have claim 10 as their base claim and are already believed allowable on this basis, but are believed further allowable for the reasons given below.

Claim 25, as amended, contains the limitation of "operating the interface whereby each of the selected instructions can be supplied *simultaneously* from the memory to the *central processing unit in a single fetch operation*", where the emphasis has been added. (Claim 10, upon which 31 and 32 depend, has a similar limitation and the "simultaneously" has been added to claim 25 to make more explicit what is meant by supplying to the processor in a single fetch.) In contrast, although Yamada can latch multiple-byte instructions at the same time, this is only *within the memory*: Yamada can only transfer data from the memory to its processing unit *one byte at a time*. This is describe at column 4, lines 16-23:

> ... data transfer occurs the same number of times as the instruction bytes contained in the cycle (i.e. data transfer is carried out every one byte of the instruction), because the data bus between the instruction processing unit and the memory has a width of one byte similar to the conventional method. For example, hen a two byte instruction is read in a cycle, data transfer is carried out twice (1 byte×2).

Consequently, as data bus 1, by which instructions are transferred from the memory to the processing unit, is only a single byte wide, multi-byte instruction *cannot*, as stated in the claims, "be supplied from the memory *to the central* processing unit *simultaneously* in a single fetch operation". Yamada illustrates and describes this with respect to Figure 4.

As shown in Yamada's Figure 4, at time T0, four data bytes are latched at the same time into the latches 200-203; however, this is only *within* the memory. When a multi-byte instruction is then transferred *out* of the memory to the processor, this is done a single byte at time. This is shown of DATA BUS 1 on the bottom line of Figure 4, where when the gate signals are used to control which bytes are transferred when. This is described in Yamada beginning at column 5, line 66. In particular, note column 6, lines 9-21 and the sequential use of signals RD0, RD1, and RD2. Consequently, the teaches of Yamada are contrary what is found in claims 25, 26, 29-32, 39, and 40, namely that "each of the selected instructions can be supplied simultaneously_from the memory to the central processing unit in a single fetch operation".

Considering the limitation of "wherein all of the selected set of instructions is stored contiguously without any gaps in the memory", the Office Action refers to column 6, lines 29-39 and 60-64. This describes addressing a three-byte instruction "when the requested three bytes are successive one ..." (column 6, line 36). In this case, the three bytes of this particular instruction will be contiguously stored without gaps, but this just shows that this can be so for a single, given instruction. Yamada does not disclose that all of the different instructions of a set of instructions are generally stored contiguously so that there are no gaps between them, but rather discusses *accessing single instructions* having contiguously addressed bytes. (Also, note Yamada states (column 6, line 36) "*when* the requested three bytes are successive one ...", which, as the added emphasis indicates does not even make clear that all of the bytes of a single multi-byte instruction are necessarily stored contiguously without any gaps.)

Consequently, it is respectfully submitted that Yamada neither teaches nor suggests that "*all* of a selected set of the instructions ... stored contiguously without any gaps"[emphasis added] in the memory.

For any of these reasons, it is respectfully submitted the Office Action's rejection of claims 25, 26, 29-32, 39, and 40 under 35 U.S.C. 102(b) as anticipated by Yamada is not well founded and should be withdrawn.

Concerning claims 30, 32, and 40, these are believed further allowable as they contain the limitation of a memory that is a one time programmable (OTP) memory. As discussed above with respect to the rejection based on Kojima, both ROM and RAM are distinct from a one time programmable (OTP) memory: a one time programmable memory allows a user to program it once, but can then not be overwritten. Yamada only refers to ROM and RAM at the locations cited in the Office Action. Consequently, a rejection of claims 30, 32, and 40 under 35 U.S.C. 102(b) is respectfully submitted to be in error and these claims are believed to further be allowable on this basis.

<u>Rejections under 35 U.S.C. 102(e) based on Suzuki *et al.*</u>

The Office Action rejected claims 28 and 38 under 35 U.S.C. 102(b) as anticipated by Suzuki *et al.*. This is respectfully submitted to be in error as the various elements of these claims are not found in Suzuki and it is believed that the Office Action is making a number of incorrect assumptions and incorrectly interpreting the teachings of Suzuki in order to fit the limitations of the claims.

First, it should be noted that claim 28 depends on claim 26, as does (through claim 27) claim 38. Claim 26 specifies that "N [a length in bytes of instructions included in the instruction set] is equal to three and M [the width in bytes of columns into which the memory is organized] is equal to four." The Office Action's rejection of is based on the values N=2 and M=3, which is contrary to the explicit limitations of the claims. Further, as far as can be determined given the length of the Suzuki patent, Suzuki neither teaches nor suggests the combination of an instruction set including 3-byte instructions (corresponding to N=3) and logically organizing a memory as a plurality of 4 byte wide columns (corresponding to M=4). Consequently, it is respectfully submitted on these bases alone that the Office Action's rejection of claims 28 and 38 under 35 U.S.C. 102(b) as anticipated by Suzuki is in error should be withdrawn. Further, in should be noted that claim 38 depends on claim 27 that also specifies the inclusion of 1-byte and 2-byte instructions.

Even aside from limitations from these limitations form the underlying claims that are not found in Suzuki, the Office Action is making a number of incorrect assumptions and incorrectly interpreting the teachings of Suzuki in order to fit the limitations of the claims. More specifically, Suzuki's Figure 13 shows a processor and compiler, where the compiler 2 stores object code in memory 130 of computer 3. As noted at column 17, lines 1-4, the memory 130, which the Office Action identifies with the memory of the claim, stores both object code and data. As for "an instruction set including N-byte instructions", the Office Action states "figure 16, N=2, see also col. 27 instructions 3 and 5, they appear to be one byte instructions as the addresses of the instructions that follow are incremented by 1". Figure 16A, as well as 16B, show examples of instruction that have been taken from the source code (1, Figure 13), compiled, and transferred to memory 130. As such, they are not just selected instructions from the instruction set, but, as is shown in Figures 16A and 16B, these compiled instruction contain addresses, register designations, flag designations, and so on in addition to the operation code specifying instructions from the instruction set; thus, even two of these complied structures are placed adjacently in the memory, their operation codes (corresponding to instructions from the instruction set) will not be adjacent and have a gap between them. The Office Action states that Figure 16A shows a N=2-byte instruction from the instruction set; rather, what it is shows is only 1-byte of operation code (including a flag designation) and 1-byte of address in a compiled instruction, as opposed to N=2-byte instruction from the instruction set.

In its rejection, the Office Action refers to "col. 27 instructions 3 and 5, they *appear to be one byte instruction*s as the addresses of the instructions that follow are incremented by 1". It is unclear what the Office Action intends to say in the portion of this statement with the added emphasis: If the Office Action means that, as the instructions following each of instructions 3 and 5 has an address incremented by one, so that therefore, they *are* 1-byte instructions, this is contrary to the Office Actions stating that Figure 16A shows an instruction with N=2. If, on the other hand, the Office Action means that, even though the address being incremented by 1 makes it appear to indicate that instructions 3 and 5 are 1-byte instructions even though they are actually N=2-byte instructions, then it appears that the Office Action is interpreting column 27 to say something it does not say. In any case, these are again compiled instructions stored in a memory for both object code and addresses, not selected elements of "an instruction set including N-byte instructions" stored in "a memory for storing the instructions". This distinction is reflected in the claim elements.

Concerning the limitation of "logically organizing the memory as a plurality rows of M byte-wide columns …", the Office Action cites column 29, line 26 of Suzuki, and states "24 bit data bus implies 3-byte columns". Although cited location does disclose a 24-bit bus, it is respectfully submitted that this implies nothing more than the bus being 3-bytes wide and to take this to imply a width for the memory is in error. Such a statement says the width of the bus between a memory and a processor determines the width into which the memory is logically organized, which is clearly not the case. For example, see the Yamada reference discussed above which has a bus between the memory and processor that is 1-byte wide, while the memory is 4-bytes wide. (Or see the rejection on page 5 of the Office Action which states Pickett logically organizes a memory to be 32-bytes wide, which does not correspond to a memory processor bus 32-bytes wide---numerous other counterexamples to this statement of the Office Action can be provided.) As far as can be determined, Suzuki neither teaches nor suggests "logically organizing the memory as a plurality rows of M byte-wide columns, wherein M is an integer greater than one and wherein N and M are relatively prime".

Concerning the limitation (as amended) of "programming a selected set of instructions of the instruction set into the memory, wherein all of the selected set of instructions is stored contiguously without any gaps in the memory", the Office Action refers to column 31 of Suzuki. At column 31, some instructions along with their addresses are shown and, judging from the addresses, it appears *some* of these may be stored contiguously; but it also appears that others, based on the gaps in the addresses, are not. As discussed above with respect to the

other references, the claim requires that *all* of the selected set of instructions be so stored. Further, there is no disclosure that these gaps are filled by instructions since, as disclosed at column 17, lines 1-4, Suzuki stores not just "object code", but data in its memory 130. Also, as discussed above with respect to Suzuki's Figures 16A and 16B, memory 130 stores compiled instructions and the memory 130 will include things such as register designations and addresses placed in between pieces of operation code. Consequently, it is respectfully submitted that Suzuki neither teaches nor suggests emphasized portion of "programming a selected set of instructions of the instruction set into the memory, wherein *all of the selected set of instructions is stored contiguously without any gaps* in the memory".

Concerning the limitation (as amended) of "operating the interface whereby each of the selected instructions can be supplied simultaneously from the memory to the central processing unit in a single fetch operation", the Office Action states "by definition a single instruction can be fetched in single instruction fetch operation". It is respectfully submitted that the Office Action is mistaken on several grounds here, namely that the this statement is incorrect, that no disclosure of such in Suzuki is presented, and that, even were this statement correct, this element of claims 28 and 38 specify more than just fetching a single instruction in a single fetch operation.

As to the Office Action's statement being incorrect, any number of processing systems uses multi-byte instructions while employing a bus of 1-byte width to transfer these instructions from the memory to the processor so that the instructions can only be transferred a byte at a time. A specific example is the Yamada reference described above: as discussed there, Yamada includes 3-byte instructions, but has only a 1-byte wide bus, so that while all three bytes of the instruction may be latched into latches *in the memory*, they are only transferred to the processor a byte at a time.

Furthermore, claims 28 and 38 specify more than just fetching a single instruction in a single fetch operation: the limitation of these claims state "each of the selected instructions can be supplied *simultaneously ... to the central processing unit* in a single fetch operation." As the added emphasis indicates, the claims state not just the instruction is fetched, but that all of it is supplied to the processor simultaneously. As far as can be determined, Suzuki neither teaches nor suggests this and the Office Action does not refer to any specific part of Suzuki's disclosure (or to Suzuki at all for this element) to provide it.

For any of these reasons, it is respectfully submitted that the Office Action's rejection of claims 28 and 38 under 35 U.S.C. 102(b) as anticipated by Suzuki is not well founded and should be withdrawn.

Conclusion

For any of the reasons given above, claims 9-12, 25-32, and 34-42 are believed allowable. Reconsideration of claims 9-12, 25-32, and 34-40, and consideration of new claims 41 and 42, are therefore respectfully requested and an early indication of their allowability is earnestly solicited.

Respectfully submitted,

Michael G. Cleveland
Reg. No. 46,030

June 27, 2005
Date

PARSONS HSUE & DE RUNTZ LLP
655 Montgomery Street, Suite 1800
San Francisco, CA 94111
(415) 318-1160 (main)
(415) 693-0194 (fax)